ISSN 2754-1118 (Online)
ISSN 2753-6599 (Print)

Enhancing Sensitivity Analysis in a Safety Case Argument

Chris Hobbs[1], Jeff Joyce[2], Julian Lapenna[3]

1 Farmhall, Ottawa, Canada.

2 Critical Systems Labs, Vancouver, Canada.

3 University of British Columbia, Canada

Abstract

A Safety Case's argument is conventionally presented as a tree with claims, strategies and evidence as nodes. The level of confidence in each node and linkage can be added and an estimate of the confidence in the top-level claim calculated. Sensitivity analysis can then highlight nodes where increased confidence would significantly improve the confidence in the top-level claim — allowing work to be focused. However, the tree structure does not show how a single factor can undermine confidence in different branches of the argument. This paper describes how adding annotations to a Safety Case can extend the sensitivity analysis to detect brittle parts of the argument.

1 Introduction

1.1 Linking Elements of the Safety Case

This paper argues that additional annotations can usefully be added to a traditional Safety Case argument to detect brittle parts of the argument. These annotations can also be used to reëvaluate the confidence structure of the Safety Case when unanticipated conditions arise.

The immediate incentive for this paper was a situation that highlighted a gap in the traditional Safety Case structure. After a Safety Case had been issued, it was found that one of the pieces of test equipment used during verification had been out of calibration. This discovery raised a doubt that cross cut much of the argument tree, compromising the confidence placed on the results of every test where that piece of equipment had been used. The Safety Case argument is traditionally represented as a tree, and this does not permit linkage between leaf nodes in different parts of the tree.

Other factors can similarly affect several parts of a Safety Case argument. For example, once a Safety Case has been issued, it may come to light that one of the verification activities was performed incorrectly and an investigation finds that the verification engineer was not properly trained, rendering the verification evidence invalid. This requires a reëvaluation of the Safety Case and it not only questions our confidence in other verifications performed by that engineer, but also in validations approved by the same assessor.

In each of these examples, the effect on the confidence of the Safety Case's top-level claim may be insignificant or catastrophic. The sensitivity analysis described in this paper is

designed to detect potentially catastrophic conditions proactively: *before the Safety Case is issued*. This could take the form of a finding such as, "*if it is found that John Smith cannot properly operate device Y, then that would cause a dramatic loss in confidence in the top-level claim*". This finding could initiate a check into John Smith's competence before any safety-affecting problem arose. If he is found to be competent, then no further action need be taken. Otherwise, another engineer should be assigned to check John's work, providing more evidence of its correctness.

In many cases, such extended sensitivity analysis exposes obvious weaknesses in the project. An extreme example could be finding that the ten verification reports created by Jill Brown were all approved by Bert Jones between 16:50 and 17:00 on a Friday afternoon. In other cases, the weaknesses are more subtle.

A Safety Case does not conventionally contain enough information to link elements of the argument for these types of correlations to be computed. This paper explores a new approach allowing such linkages to be made. If adopted within a development organisation, it would enable the company to focus and prioritise resources towards the "brittler" areas of its Safety Cases; ensuring that the areas of most consequence are appropriately addressed.

1.2 The Structure of this Paper

Section 2 describes the annotation commonly used to express levels of confidence in a Safety Case argument: the Goal-Structuring Notation (GSN) with Bayesian extensions. Although the GSN is well-known, the Bayesian extensions are less familiar and, as the technique introduced in this paper depends heavily on those extensions, this section provides a short tutorial.

The method proposed in this paper is inspired by the Functional Resonance Analysis Method (FRAM). Section 3 provides a brief introduction to FRAM and the application of FRAM's resonance technique to a Safety Case.

In order to apply the proposed method, additional influences that cut across the tree structure have to be added to the Safety Case. Section 4 describes an example and the results obtained from that example.

Section 6 summarises the results from the example, and Section 5 addresses the question of how the technique could be applied in practice.

2 Confidence in a Safety Case

2.1 Types of Confidence

GSN, as defined by the Safety-Critical Systems Club (2021), defines a format for structuring the argument and evidence to support a particular claim — often that a product is sufficiently safe or sufficiently secure. However, basic GSN does *not* provide a means to assign levels of confidence to the elements of the Safety Case to specify:

- how much confidence we have in the claim or evidence itself, and
- how strongly we believe that a claim or evidence supports the argument.

For example, we may provide both a formal proof and the results of dynamic testing as independent pieces of evidence for the correctness of a complex algorithm. If correct, the formal proof would provide stronger support for the claim than the test results, but we may not yet have had the proof fully reviewed by an expert. Although we may be fully confident in the correctness of the test results, we might assign a fairly low confidence level to how the test results support the claim because it is unlikely that any finite amount of testing would cover the algorithm's state space.

This example is captured in Figure 1, where the two types of confidence are estimated:

1. We are very confident (90%) that the dynamic test results have been correctly executed and recorded, but we are not confident (17%) that, on their own, they provide much support to the claim that the algorithm is correct.
2. Because it has not yet been checked, we are only 50% confident that the formal proof is correct, but we are 100% confident that, if correct, it would strongly support the claim, even if no other evidence were available.

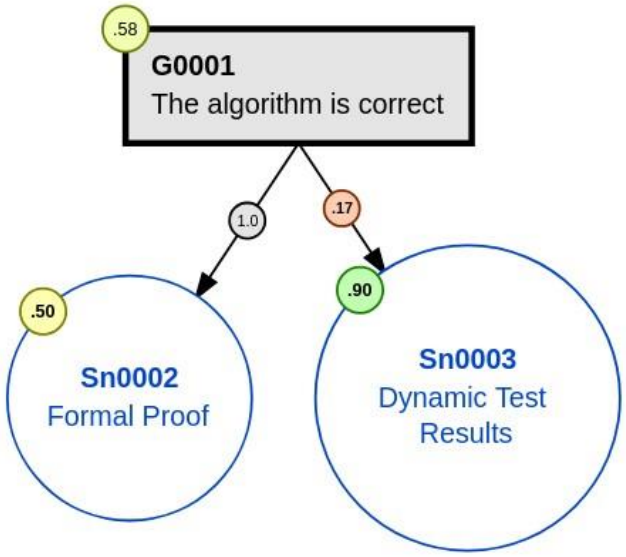

Figure 1 ~ Confidence Levels: Noisy-OR

2.2 Sensitivity Analysis

The two types of confidence in the example above can be added to Safety Case arguments using different annotations. This paper uses Bayesian annotations, but the technique could equally well apply to Dempster-Shafer annotations (Dempster–Shafer theory n.d.).

The use of Bayesian Belief networks to quantify confidence is an area of growing focus in safety-critical industries. Recent literature has explored applications of this, including confidence propagation based on Dempster-Shafer theory (Guiochet et al. 2019), formal methods for defining belief within GSN-to-BBN transformations (Nešić et al. 2021), the application of BBNs to formalise dynamic, through-life safety cases (Denney et al. 2015) and preparing machine-processable, mathematically sound but still human-understandable models of complex systems within road vehicles (Maier et al. 2024). Hobbs and Lloyd (2012), also Fenton and Neil (2013), contain examples of the argument structures as represented by a Bayesian Belief Network (BBN).

Once such quantification is in place, it is possible to perform a sensitivity analysis to see where work should be focussed. In the example in Figure 1 it would clearly be more profitable to spend time increasing our confidence in the formal proof than in doing more work on the dynamic tests.

2.3 Noisy Conjunctions

For each node, the contributions of the child nodes are calculated using a Noisy-OR or Noisy-AND conjunction (Fenton and Neil 2013). These noisy conjunctions are more expressive than the Boolean equivalents. Noisy-OR, for example, allows us to say that Y is true if x_1 OR x_2 OR … OR x_n is to some extent true, but Y may also be true if all the x_i are false. This allows an expression of the form, "*The patient is in danger if the wrong drug is given OR if the wrong dose is given*" while recognising that the wrong drug or wrong dose does not *always* cause injury, and that there may be other factors that could also put the patient in danger, e.g. an unsterilised needle.

The two pieces of evidence in Figure 1 are combined by a Noisy-OR as each independently supports the claim.

The Noisy-OR function calculates the overall confidence level as:

$$P\left(\mathrm{T=true}\middle|x_1,x_2,\ldots,x_n\right) = 1-(1-k)\times\prod_{i=1}^{n}(1-x_i v_i)$$

where v_i is the link strength and k is the "leakage": the confidence we have in the claim, independent of the pieces of evidence.

Analogous to Noisy-OR, a BBN also offers Noisy-AND. Figure 2 argues that, in order to claim that the system's hazards have been adequately mitigated, it is necessary that all hazards have been identified AND that each has been mitigated. Again, as with the Noisy-OR, a leakage can be specified: the lack of confidence in the top-level even where there is total confidence in all the subclaims. In Figure 2 the leakage, k, is set to 0.1.

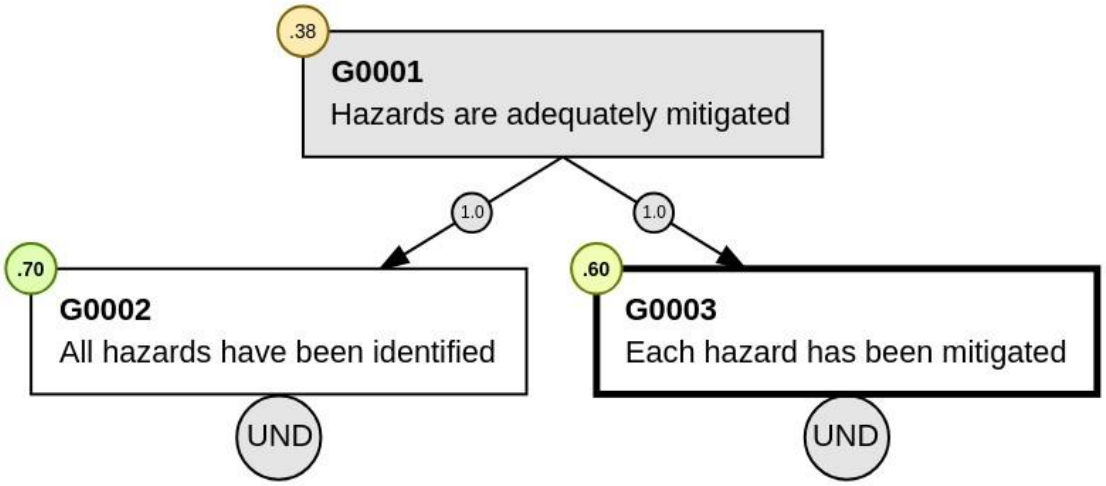

Figure 2 ~ Confidence Levels: Noisy-AND

The Noisy-AND function calculates the overall confidence level as:

$$P\left(\text{T=true}\middle|x_1,\text{x}_2,\ldots,\text{x}_n\right) = (1-k) \times \prod_{\text{i=1}}^{n}(1-(1-x_i)v_i)$$

Typically, a Noisy-OR starts out with high confidence that reduces during analysis, while a Noisy-AND starts with low confidence which increases during analysis.

2.4 Eliciting Confidence Levels

Two criticisms are often levelled at quantifying confidence in a Safety Case:

1. Obtaining consistent and accurate confidence levels from experts is not simple, whether for inclusion in a Safety Case or a failure analysis. O'Hagen et al. (2006) present a methodological framework for eliciting such expert knowledge based on findings from statistical and psychological research, but even there it is acknowledged that results may not always be accurate.
2. It is very easy for a BBN tool to combine several estimates of the type "about 80%, about 85%, about 70%, …" into a precise but spurious confidence distribution with a mean of 73.3662625% rather than the more honest "about 75%".

The approach proposed in this paper avoids these criticisms by accepting that confidence levels as given in the Safety Case are starting points to be manipulated rather than accurate values, and by dealing with ratios of confidence values rather than absolute values.

3 Resonance

3.1 Functional Resonance Analysis Method (FRAM)

Hollnagel (2012) describes functional (as opposed to stochastic) resonance: how, in a socio-technical system, the variability caused by humans making adjustments based on what others might do can result in non-linear, emergent behaviour.

A FRAM analysis recognises and models the dependencies between system functions that result in the performance variabilities of the functions becoming coupled.

Standard Safety Case notations don't allow the linkages between nodes in the argument structure to be represented. By adding these, a sensitivity analysis can make small alterations to specific aspects, e.g. John Smith's competency, and determine whether these would lead to unexpected emergent behaviour.

3.2 The Dynamic Safety Case

Producing a project's Safety Case has traditionally been seen as a "wrapping up" activity towards the end of the project. Hobbs et al. (2024) argue that advantages can be gained by creating the Safety Case early in the project, and give an example of a project where doing so actually saved work. In this case the Safety Case becomes more dynamic, but still tends to be frozen once the product is certified and shipped.

Many systems currently being deployed will meet situations that were not considered during the development of the Safety Case and, in this case, the Safety Case needs to be

dynamic: to consider the difference to the Safety Case resulting from the new circumstances (Diemert et al 2023). This is particularly valid for autonomous systems where events will be met that were not part of the training data (see Koopman (2018)), but is also true of non-autonomous systems that are intended to have a long life.

Additionally, Fenn et al (2025) point out that a Safety Case created during a product's design and implementation often implicitly relies on the product being adequately maintained after shipment, something that may not happen. That paper extends the Safety Case argument to include dynamic maintenance data once the product is in the field.

The impact analyses carried out during the deployed life of the product, particularly those monitoring Safety Performance Indicators (SPIs), will change our level of confidence in the claims and evidence as new situations arise. Where there are hidden linkages between elements in the Safety Case, resonance may occur.

For example, we may have high confidence in a claim that an autonomous car can react correctly to traffic lights, based on the evidence of many hours of driving in the USA and Europe. That confidence could be undermined when the car first meets a horizontal Québec traffic light with square lights, or travels behind a truck carrying traffic lights.

This does not change the structure of the Safety Case argument: it simply reduces our confidence in the evidence, because we realise that it is incomplete.

4 Example

4.1 System Description

It was difficult to find a system which was sufficiently complex to represent a real-world safety argument, while being sufficiently compact to describe in this paper. An example was chosen from the railway industry: see Figure 3.

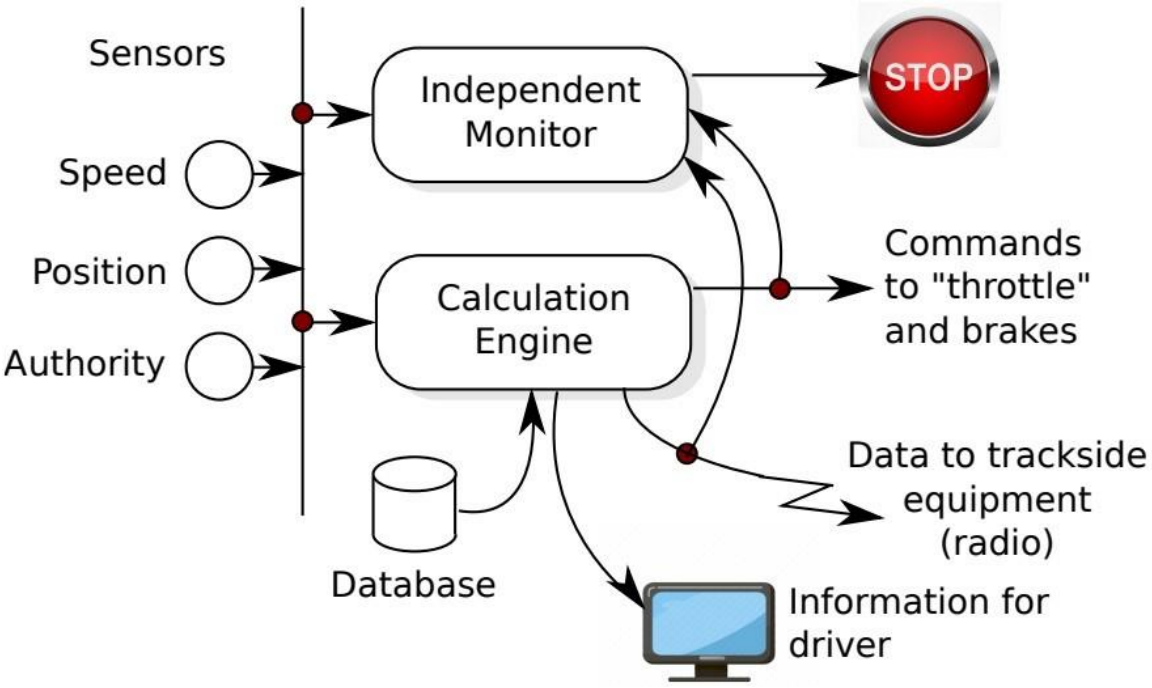

Figure 3 ~ Example System

This system reads the current speed (downward facing LIDAR) and position (GNSS) of the train, and its most recent movement authority (from a balise). It performs the necessary calculations, given the gradient and curvature of the track ahead, displays this information to the human driver and, as appropriate, automatically applies power or brakes. It also sends position and speed information to the trackside equipment.

Given the complexity of this computation, an independent monitor is applied (called a "*safety bag*" in EN 50716:2023) which reads the sensor inputs and the computational results. It performs a much coarser calculation than the calculation engine, but if it detects an unsafe command to the throttle or brakes, it applies the brakes and brings the train to a halt.

The safety constraints on this system are very easy to meet: if the train is never allowed to move, all of the safety constraints will be met. However, it is also essential that the train be useful, and that naturally reduces its safety. The independent monitor commanding the train to stop when the calculation engine produces unsafe output may keep the train safe, but by stopping on the tracks, the train becomes a hazard in the larger railway system: another train, exceeding its movement authority, might run into it. Therefore there are requirements both on the safety of the train (driven by the independent monitor) and on the availability and reliability of the calculation engine. The Safety Case argument must take these conflicting requirements into account.

4.2 Confidence Influences

Table 1 lists the aspects of the evidence nodes included in the analysis of the example. In many cases the same aspect was used several times on the same node: there were perhaps two approvers of a document, or several pieces of equipment used in a test. These aspects were selected because they represent common cross-cutting patterns observed across a wide range of safety-related projects: shared personnel, tools or review processes. The list of such aspects is potentially endless, particularly if the supply chain is included: should the supplier of each piece of equipment be included in case there is a flaw in its quality control system? And the supplier of the components for that equipment?

Table 1 ~ System Aspects Added to Safety Case

Aspect	Meaning	Format
Prime	Person responsible for producing the evidence	<unique id (name)>
Approver	Person who approved the evidence	<unique id (name)>
Equipment	Equipment used	<serial number>
Review Type	Whether confirmation review or not (if no Review Type, then assumed to be FULL)	<CONF or FULL>
Approval DT	UTC Date/time of approval	<yyyymmddhhmmss>
Submission DT	UTC Date/time of submission	<yyyymmddhhmmss>
Type	Formal, Semiformal or Informal	<F, S or I>
Pages	Number pages in a document	<integer>

4.3 Example Safety Argument

The Safety Case argument for the example system was created using the Socrates tool from Critical Systems Labs (CSL) and has 58 claim (goal) nodes, 5 strategy nodes and 42 evidence nodes. The top-level claim is based on the strategy applied to the Open Autonomy Safety Case (Wagner and Carlan 2024):

1. the system was developed in an environment with a strong safety culture ("*we live it right*").
2. the system was designed and built with a high level of rigour ("*we engineer it right*").
3. once delivered to a customer, the system is monitored and the SPIs tracked ("*we operate it right*").

This approach ensured that the Safety Case used in the experiment was genuine but not specifically written with the experiment in mind. The top-level claims are illustrated in Figure 4 and a static report of the entire Safety Case can be found at https://scsc.uk/documents/ejournal/network.pdf.[1]

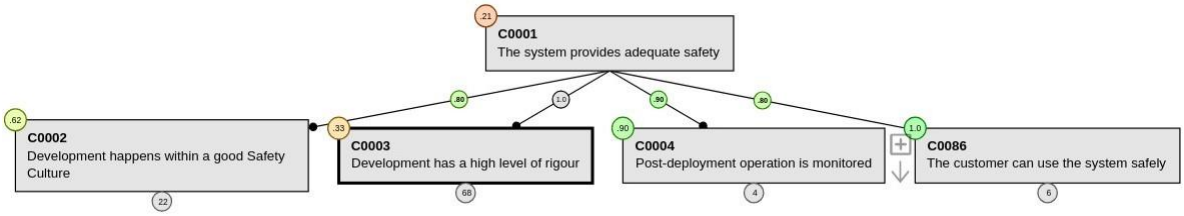

Figure 4 ~ The Example's Top-Level Claim (Goal)

4.4 Method Used in the Example

The Socrates tool used to enter the example Safety Case provides the means to carry out a sensitivity analysis, but not an analysis of the system aspects. For the purposes of this paper, the Safety Case was exported from Socrates in JSON[2] format and manipulated by an external Python program.

4.5 Results from the Example

Given the aspects stored with each evidence node, it is possible to search for many patterns that could seriously affect the confidence in the top-level claim. A pattern might be the same piece of test equipment being used in many verification activities and the calculation that, if this were out of calibration, confidence in the top-level claim would be significantly reduced. This could lead to a recommendation to check the calibration on that piece of equipment.

For the example Safety Case, the patterns sought included the following:

[1] This QR code provides the same Uniform Resource Locator (URL) for those with a printed copy:

[2] JavaScript Object Notation

1. An approver who has approved a report or analysis in an unreasonably short time. This depends on the type and size of the report and on whether the approver is assessing the content of the document or is simply confirming that the correct process has been followed in completing and reviewing it (a confirmation review).
2. A document that has never been approved. In general, it is assumed in the example that meeting minutes are not approved, but that design and verification reports and analyses do require approval.
3. An approver who has approved numerous reports or analyses within an unreasonably short time.

These conditions throw doubt onto other actions by the same person, and so the confidence associated with those is reduced and the overall confidence regenerated.

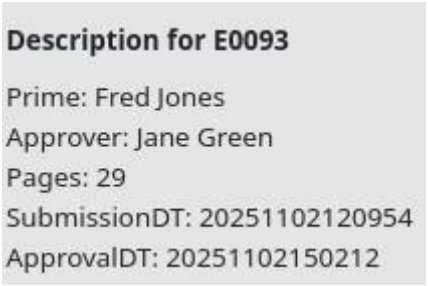

Figure 5 ~ Example Aspects

Figure 5 gives an example of the aspects being added to a node: in this case to the verification report to support the claim that the availability requirements have been met. It can be seen that Jane approved Fred's report just under three hours after it was made available. There is no indication in Figure 5 that this was a confirmation review and the program assumed that reviewing the contents of a 29-page document would take longer than this and displayed a warning:

> Verification report VE001998 was approved at 2025-11-02T15:02:12 by Jane Green after submission at 2025-11-02T12:09:54. Elapsed time was 2 hrs 52 mins.

The program then looked for other places where Jane Green had performed a review and checked to see how doubting those reviews would affect the confidence in the top-level claim. In the case of Jane Green, the change in confidence is significant because she had approved almost all of the verification reports:

> Casting doubt on approvals by Jane Green reduces confidence in top-level claim by 67.68%

During its analysis of the Safety Case, the program identified a number of such points of weakness.

5 The Practicality of this Approach

Applying the technique proposed in this paper means that aspects must be added at least to the evidence nodes in the Safety Case. In preparing the example described in Section 4, this manual exercise was found to be tedious. Adding dates and times was particularly tiresome.

However, in a real-world situation, the aspects would be URLs linking to the actual piece of evidence (known as “artifacts” in the Socrates tool). These URLs would typically link

into a configuration control system where the size of the document, the names of the prime and approver and the dates and times of submissions would be available.

The technique would therefore prove more practical in real-world applications, where it would integrate naturally with automated data capture processes and configuration management and verification tools.

Rather than producing a program to scan a textual representation of the Safety Case argument, as was performed for this paper, it could also be possible to use a Large Language Model (LLM) to perform the scanning, looking for the potentially dangerous conditions.

6 Summary and Further Work

A traditional sensitivity analysis on a Safety Case identifies where it would be worth spending more time to increase the confidence in a particular part of the argument. But the tree structure of the Safety Case argument makes it difficult to determine how different branches of the argument affect one another. The technique proposed in this paper is to extract additional information from the evidence nodes by following their embedded links to the documents or equipment data in the company's configuration control system.

This allows commonalities (same approver, same equipment, etc.) between nodes in different branches of the tree to be identified. A sensitivity analysis can be more accurate as it modifies nodes in different branches of the tree simultaneously.

Applying the proposed technique to the artificial, but realistic, example argument in Section 4 indicates that it is able to identify areas of weakness in a Safety Case. In the example Safety Case, cross-branch dependencies were detected that reduced confidence by up to 68%. This gap in approaches to traditional Safety Case demonstrates the need for proactive identification of cross-cutting weaknesses before certification.

Of course, unless a company is working within a good safety culture, the checks can be easily circumvented! "*I won't bother reading your analysis, but I'll let it sit for a couple of days before I approve it so that I'm not pulled up by the Safety Case tool*."

Further work is needed to apply this technique to a real-world Safety Case, particularly one larger than the 105-node example.

Acknowledgments

We would like to thank one of the anonymous reviewers of the first draft of this paper, for his or her very useful and insightful comments.

References

Dempster–Shafer theory. (no date). In Wikipedia: https://en.wikipedia.org/wiki/Dempster–Shafer_theory. Accessed 8th January 2026.

Denny E, Habli I, and Pai G. (2015). *Dynamic Safety Cases for Through-Life Safety Assurance*. ICSE '15: Proceedings of the 37th International Conference on Software Engineering – Volume 2, Pages 587–590.

Diemert S, Goodenough J B, Joyce J, and Weinstock C B. (2023). *Incremental Assurance Through Eliminative Argumentation*. Journal of System Safety, Volume 58, Number 1.

EN 50716. (2023). *Railway Applications - Requirements for software development*. EN 50716, 1st Edition, 2023. CENELEC (European Committee for Electrotechnical Standardization), Brussels.

Fenn J, Hawkins R D, and Nicholson M. (2025). *Practical Examples of a New Approach to Creating Clear Operational Safety Cases*. In: Parsons M. (editor) *Developing Safer Complex Systems: Proceedings of the 33rd Safety-Critical Systems Symposium, York, UK, 4–6th February 2025*. Safety Critical Systems Club.

Fenton N, and Neil M. (2013). *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, Boca Raton.

Guiochet J, Wang R, Motet G, and Schön W. (2019). *Safety Case Confidence Propagation Based on Dempster-Shafer Theory*. International Journal of Approximate Reasoning, Volume 107, April 2019, Pages 46–64.

Hobbs C, and Lloyd M. (2012). *The Application of Bayesian Belief Networks to Assurance Case Preparation*. In: Dale C, Anderson T. (editors) *Achieving Systems Safety: Proceedings of the Twentieth Safety-Critical Systems Symposium, Bristol, UK, 7–9th February 2012*. Springer, London

Hobbs C, Diemert S, and Joyce J. (2024). *Driving the Development Process from the Safety Case*. In: Parsons M. (editor) *Safe AI Systems: Proceedings of the 32nd Safety-Critical Systems Symposium, Bristol, UK, 13–15th February 2024*. Safety-Critical Systems Club.

Hollnagel E. (2012). *FRAM: The Functional Resonance Analysis Method — Modelling Complex Socio-technical Systems*. CRC Press, Boca Raton.

Koopman P. (2018). *The Heavy Tail Safety Ceiling*. Automated and Connected Vehicle Systems Testing Symposium, Greenville, South Carolina, USA.

Maier R, Grabinger L, Urlhart D, and Mottok J. (2024). *Causal Models to Support Scenario-Based Testing of ADAS*. IEEE Transactions on Intelligent Transportation Systems, Volume 25, Number 2, February 2024, Pages 1815–1831.

Nešić D, Nyberg M, and Gallina B. (2021). *A probabilistic model of belief in safety cases*. Safety Science, Volume 138, June 2021.

O'Hagan A, Buck C E, Daneshkhah A, Eiser J R, Garthwaite P H, Jenkinson D J, Oakley J E, and Rakow T. (2006). *Uncertain Judgements: Eliciting Experts' Probabilities*. John Wiley & Sons, Ltd, Chichester.

Safety-Critical Systems Club. (2021). *GSN Community Standard*. Technical Report SCSC-141C, Safety-Critical Systems Club. https://scsc.uk/r141C. Accessed 8th January 2026.

Wagner M, and Carlan C. (2024). *The Open Autonomy Safety Case Framework*. Safety-Critical Systems eJournal, Volume 3, Issue 1. https://scsc.uk/r1939. Accessed 8th January 2026.

This collation page left blank intentionally.